US-PAT-NO: 5867602

DOCUMENT-IDENTIFIER: US 5867602 A

TITLE: Reversible wavelet transform and embedded codestream manipulation

DATE-ISSUED: February 2, 1999

INVENTOR-INFORMATION:

NAME	STATE	ZIP CODE	COUNTRY	CITY
Zandi; Ahmad	CA	N/A	N/A	Cupertino
Schwartz; Edward L.	CA	N/A	N/A	Sunnyvale
Gormish; Michael J.	CA	N/A	N/A	Los Altos
Boliek; Martin	CA	N/A	N/A	San Francisco

US-CL-CURRENT: 382/248, 382/233 , 382/240

ABSTRACT:

A compression and decompression system in which a reversible wavelet filter are used to generates coefficients from input data such as image data. The reversible wavelet filter is an efficient transform implemented with integer arithmetic that has exact reconstruction. The present invention uses the reversible wavelet filter in a lossless system (or lossy system) in which an embedded codestream is generated from the coefficients produced by the filter.

An entropy coder performs entropy coding on the embedded codestream to produce the compressed data stream.

72 Claims, 45 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 30

---------- KWIC ---------

Detailed Description Text - DETX (7):
bit-significance: A number representation, similar to sign magnitude, with **head bits,** followed by the sign **bit,** followed by **tail bits,** if any. The embedding encodes in bit-plane order with respect to this representation.

Detailed Description Text - DETX (38):
In one embodiment, the ordering/modeling block 103 comprises a sign/magnitude formatting unit 201 and a joint space/frequency **context model** 202, such as shown in FIG. 2. In one embodiment, the joint space/frequency **context model** 202 comprises a horizon **context model,** as is described below. The input of the sign/magnitude unit 201 is coupled to the output of the **wavelet** transform coding block 102. The output of sign/magnitude unit 201 is coupled to joint space/frequency modeling block 202. The output of JSF **context model** 202 is coupled to the input of entropy coder 104 which produces the

output code stream 107.

Detailed Description Text - DETX (134):
There are three types of **bits** in a number represented in **bit**-significance form: **head, tail,** and sign. The head bits are all the zero bits from the MSB to the first non-zero magnitude bit plus the first non-zero bit. The bit-plane where the first non-zero magnitude bit occurs defines the significance of the coefficient. The bits after the first non-zero magnitude bit to the LSB are the tail bits. The sign bit simply denotes the sign. A number with a non-zero bit as the MSB has only one head bit. A zero coefficient has no tail or sign bits.

Detailed Description Text - DETX (165):
The present invention may be implemented in hardware and/or software. A hardware implementation of the present invention requires implementation of the **wavelet** filters, memory/data flow management to provide the data for the filters, a **context model** to control the embedded coding of the present invention, memory/data flow management to provide the data for the **context model,** and a binary entropy coder.

Detailed Description Text - DETX (194):
In one embodiment, the horizon **context model** of the present invention comprises the bit-significance embedded encoding of the **wavelet** coefficients

that feeds a binary entropy coder.

Detailed Description Text - DETX (210):

In one embodiment, there is no coding until the first one bit. At the occurrence of the first one bit in the coefficient, the sign is encoded. Although the **head bits** are image/region dependent, the **tail bits** are more uniform across different images and regions. Based on how far the **tail bits** are from the initial one **bit (in the head bit**), certain probability classes are used to encode the **bits in the tail**. In one embodiment, the first tail bit in a coefficient is coded with a probability class including 0.7. The second and third tail bits are coded with a probability class including 0.6. Lastly, the fourth and further tail bits are coded with probability classes that includes 0.5.

Detailed Description Text - DETX (240):

In order to use normalized filters, an alignment unit between the forward **wavelet** filter 1600 and the **context model** 105, can be used to compensate for the energy gained (or alternatively, lost) from the unnormalized filter, which improves compression. Because alignment allows non-uniform quantization for lossy operation, alignment can enhance the visual quality of lossy image reconstructions. In the one-dimensional case, coefficients from each level of the tree would have different alignment (divisors=.sqroot.2, 2, 2.sqroot.2, 4,

multipliers =2.sqroot.2, 2, .sqroot.2, 1). In the two-dimensional case, the divisors would be 2, 4, 8, 16 and the multipliers would be 8, 4, 2, 1.

Detailed Description Text - DETX (258):

If the **tail**-on **bit** of the present coefficient is zero (for **head bits**), then 1024 contexts from the **tail**-information **bits** of the parent and W coefficient and the **tail**-on **bit** of the NW, N, NE, E, SW, and S coefficients respectively. In one embodiment, adaptive coding is used for head bits. In some embodiments, a single context is used to provide some "run coding" of head bits. If the next 16 **bits** to be coded are all **head bits** and their N, S, E, and W neighbors and parent all have **tail**-information 0, a single decision will be coded. This decision indicates if any of the 16 bits to be coded has a one bit at the current bitplane. If there is no one bit, the 16 decisions normally coded can be skipped. If any of the next 16 coefficients contain their first significant bit, then 16 decisions are used one for each bit. This "look ahead" results in fewer calls to the binary entropy coder which results in higher speed and higher compression.

Detailed Description Text - DETX (278):

The context model of the present invention is shown in block diagram form in FIG. 27. Context model 2700 contains the sign/magnitude unit 109 (FIG. 2), and three units for processing the different bits in

the coefficient. Based on the bit being coded, one of the three units is selected. A switch may be included to facilitate the switching between the units in a hardware implementation. These units include a **head bit** block 2701, a sign **bit** block 2702, and a **tail bit** block 2703. The **head bit** block 2701, a sign **bit** block 2702, and a **tail bit** block 2703 model the **head bits,** the sign and the **tail bits,** respectively, as described above. The output of these three units is sent to the entropy coder 104 (FIG. 1).

Detailed Description Text - DETX (280):
The contexts defined above are used with an adaptive binary entropy coder with a few exceptions. The contexts of the **head bits** (present coefficient **tail**-on **bit**=0) and the sign **bits** when N.sub.--**tail**-on=1 are allowed to adapt.

Detailed Description Text - DETX (351):
FIG. 30 illustrates a system utilizing a channel manager. Referring to FIG. 30, **wavelet** transform 3001 generates coefficients. These coefficients are subjected to **context model** 3002. **Context model** 3002 is coupled to a channel manager 3003 that includes a buffer memory. The channel manager 3003 is coupled to a limited bandwidth channel 3004.

Detailed Description Text - DETX (394):
This is important because compression and processing causes images to drift

farther from the original. If the compressor is idempotent, then multiple lossy compression decompression cycles do not affect the data. In the present invention, it does not matter how many times data is compressed and decompressed at the same compression ratio. Also, a lossy input to a parser subjected to further quantization produces an identical result to the case when a lossless input is used. Thus, the present invention comprises a transform-based idempotent system that includes a **wavelet** transform, a **context model,** and an entropy coder, such that coefficients are described and stored in an order such that removing information does not change the description for prior coefficients.

Detailed Description Text - DETX (411):

The present invention is relatively simple to implement in both software and hardware. The **wavelet** transform can be calculated with just four add/subtract operations and a few shifts for each high-pass, low-pass coefficient pair. The embedding and encoding is performed with a simple "**context model**" and a binary "entropy coder". The entropy coder can be performed with a finite state machine or parallel coders.

Claims Text - CLTX (21):

18. The encoder defined in claim 1 wherein the ordering and modeling mechanism comprises a context model that codes a single decision if a plurality

of **head bits** are to coded and their north, south, east, and west neighbors and parents all have **tail**-information 0.

Claims Text - CLTX (70):

56. The apparatus defined in claim 46 wherein the means for ordering comprises a context model that codes a single decision if a plurality of **head bits** are to coded and their north, south, east, and west neighbors and parents all have **tail**-information 0.

Claims Text - CLTX (85):

68. The decoder defined in claim 62 wherein the model and binary entropy coder decode a single decision if a plurality of **head bits** are to coded and their north, south, east, and west neighbors and parents all have **tail**-information 0.

PUB-NO: EP000914004A1

DOCUMENT-IDENTIFIER: EP 914004 A1

TITLE: Coding system and method for lossless and lossy compression of still and motion images

PUBN-DATE: May 6, 1999

INVENTOR-INFORMATION:

NAME	COUNTRY
WU, XIAOLIN	CA
BARTHEL, KAI UWE	DE

ASSIGNEE-INFORMATION:

NAME	COUNTRY
NTEC MEDIA GMBH	DE

APPL-NO: EP97118846

APPL-DATE: October 29, 1997

PRIORITY-DATA: EP97118846A (October 29, 1997)

INT-CL (IPC): H04N007/26

EUR-CL (EPC): H04N007/26 ; H04N007/26, H04N007/26 , H04N007/26

ABSTRACT:

CHG DATE=19990702 STATUS=O> The invention relates to a coding system for lossless and lossy compression of still and motion images with means of statistical **context modeling** for adaptive entropy coding of **wavelet** coefficients in different conditioning templates. Furthermore, it is directed to a method for lossless and lossy compression of still and motion image data by hierarchical decomposing of said image data into subbands by a revertible **wavelet** transform that generates **wavelet** coefficients and by using a conditioning template for statistical **context modeling** and adaptive entropy coding of said **wavelet** coefficients. The compression ratio is improved by one or more of the following steps: adapting the shape and/or orientation of said conditioning template to different subbands; converting a two-dimensional array of signed **wavelet** coefficients into an equivalent sequence of only two input symbols for adaptive binary entropy coding; using previously scanned bit planes in forming conditioning templates; reducing the number of possible conditioning states corresponding to all possible combinations of events in the conditioning templates by using least-squares estimates of magnitudes of **wavelet** coefficients; reducing the number of possible conditioning states corresponding to all possible combinations of events in the conditioning templates by first using least-squares estimates of magnitudes of

wavelet conditions: and then by minimum-entropy quantization of said estimates: comparing the so-far-coded bits of the coefficient @ being presently coded with the so-far-coded bits of the neighbouring coefficients and parent coefficient of @ to characterise spacial texture patterns and using them to augment the conditioning states created by the quantization of said estimates; conditioning the sign of a **wavelet** coefficient @ on the signs of neighbouring coefficients of @ . recording for each subband the location of the **most significant bit** of the coefficient of maximum magnitude in the subband and including it as side information in the code stream.

US-PAT-NO: 5091955

DOCUMENT-IDENTIFIER: US 5091955 A

TITLE: Voice coding/decoding system having selected coders and entropy coders

DATE-ISSUED: February 25, 1992

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Iseda; Kohei	Kawasaki	N/A	N/A	JP
Satoh; Kazumi	Yokohama	N/A	N/A	JP
Kurihara; Hideaki	Kawasaki	N/A	N/A	JP
Amano; Fumio	Tokyo	N/A	N/A	JP
Unagami; Shigeyuki	Atsugi	N/A	N/A	JP
Okazaki; Koji	Kawasaki	N/A	N/A	JP

US-CL-CURRENT: 704/230, 704/221

ABSTRACT:

Disclosed is a voice coding/decoding system having a transmitting part for transmitting a coded signal of an input voice signal at a bit rate lower than a predetermined transmission bit rate and a receiving part for receiving and decoding the coded signal transmitted from the transmission part. To enable

the coding and transmitting of an input voice signal in an optimum state without passing through a buffer memory and without having a negative influence on the coder, the transmitting part provides coders for coding the input voice signal and groups of entropy coders. The inputs of the entropy coders in each group are connected to the output of one of the plurality of coders. The transmitting part further provides an evaluation part for evaluating the characteristics of the outputs of the coders and the entropy coders. The evaluation part extracts those entropy coders having output bit rates lower than the transmission bit rate and extracts, from the coders connected to the extracted entropy coders, a coder having the best output characteristic. Then, the evaluation part outputs a selecting signal indicating the combination of the selected coder and an entropy coder from the extracted entropy coders. The transmitting part further provides a selecting part for selecting, in response to the selecting signal, the codeword passed through the combination of the coder and the entropy coder to be transmitted.

59 Claims, 7 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 6

---------- KWIC ---------

Brief Summary Text - BSTX (10):

Conventionally, to solve the above problem, the entropy encoded result is stored in a buffer memory and is transmitted from the buffer memory through the transmission line. To this end, a buffer control system has been employed in which the entropy-coded results are controlled by changing the quantization characteristics of a quantizer in the coder. This conventional system is described in "On the Information **Rate Control** in Entropy-coded Speech Transmission Systems", written by M. Copperi, CSELT Rapportitecnici Vol. X-No. 6-DECEMBER, 1982 PP 435-449.

Detailed Description Text - DETX (3):
As described before, since the voice signals have nonuniform probability of symbols, the statistical characteristics of the output of the coder are changed so that the code lengths, i.e., bit rates, of the entropy-coded results are not constant, and the bit rate may be larger than the transmission bit rate so that the transmission becomes impossible. To solve this problem, the entropy-coded result is stored in the buffer memory 73 and is transmitted from the buffer memory 73 to the transmission line. To this end, a buffer control system has been employed in which the entropy-coded results are controlled by changing the quantization characteristics of a quantizer in the coder. This conventional system is described in "On the Information **Rate Control** in Entropy-coded Transmission Systems", written by M. Copperi, CSELT Rapportitecnici Vol. X-No.

6-DECEMBER, 1982 PP 435-449.

Detailed Description Text - DETX (53):

In this embodiment, two paths without passing through an entropy coder are provided as mentioned before. This is to ensure that the codeword can be surely obtained even when all of the bit rates of the codewords from the H coders 55 and 56 exceed the transmission bit rate of 4 bits/sample in the case when the assumed probability distribution of the codeword is greatly different from the actual distribution. It is always possible to transmit the output of the 4-bit ADPCM coder. In other words, when the assumed probability distribution of the **codeword** is greatly different from the actual probability distribution, the **average codeword length** of the entropy-coded words greatly exceeds the **codeword length of a codeword** before the entropy codings. In such a case as above, the entropy coding is not employed. This corresponds to the case when entropy coders of uniform characteristics are combined.

US-PAT-NO: 6229927

DOCUMENT-IDENTIFIER: US 6229927 B1

TITLE: Reversible embedded wavelet system implementation

DATE-ISSUED: May 8, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Schwartz; Edward L.	Sunnyvale	CA	N/A	N/A

US-CL-CURRENT: 382/248, 382/240 , 382/244

ABSTRACT:

A method and apparatus for performing compression and/or decompression is described. In one embodiment, the present invention comprises a system having a buffer, a wavelet transform unit, and a coder. The wavelet transform unit has an input coupled to the buffer to perform a wavelet transform on pixels stored therein and to generate coefficients at an output. The coder is coupled to the wavelet transform unit to code the transformed pixels received from the buffer.

13 Claims, 58 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 46

---------- KWIC ---------

Detailed Description Text - DETX (377):
Encoder **Rate Control**

Detailed Description Text - DETX (378):
In addition to having the ability to quantize data, performing **rate control** in the encoder also requires measuring the rate so that decisions on quantization can be made. If the rate indicates that compression is not good (i.e., not at a desired level), quantization may be increased. On the other hand, if the rate indicates that compression is too high, quantization may be decreased. **Rate control** decisions must be made identically in the encoder and the decoder.

Detailed Description Text - DETX (382):
Another option for rate management illustrated as the smaller circle (position 2 in FIG. 34) is to count the start of interleaved words in the encoder. In another embodiment, this is performed after the bit generation stage (position 4 in FIG. 34). Because the encoder and decoder start a codeword at the same time, implicit signaling of the rate may be used. The counting may be performed with counting hardware that comprises a register and an adder that adds the **codeword lengths** and determines the **average codeword**

length. Hardware to perform the counting and determining average numbers of bits is well-known in the art and is shown in FIG. 34 as block 3401. It would be apparent that this block may be used to take similar measurements at other locations in the system (e.g., positions 1, 2, 3, 4, on both encoder and decoder).

Detailed Description Text - DETX (384):

Rate measurement can be implicit: both the encoder and decoder perform the same rate determination calculation. For example, the encoder and decoder could accumulate the average size of a codeword each time a new codeword is started. This is represented by position 4 in FIG. 34. (The actual size cannot be used, since the encoder does not know the size until the end of the codeword). If the R-codes used in the core vary in size from R2(0) through R2(7), the average codeword size varies from 1 to 4.5 bits. If probability estimation works well, using the average should be very accurate. In other cases, the differences between the minimum and the maximum **codeword lengths** versus the **average** are typically not so great, so the estimate should still be useful. The average size of a Rz(k) codeword is k/2+1 bits.

Detailed Description Text - DETX (386):

Another benefit of encoder **rate control** is that the encoding of less important data can be stopped when the maximum

bandwidth is exceeded. This
increases the speed of encoding, and decreases the
total time to output data
(e.g., decrease the total time to print).

US-PAT-NO: 5815097

DOCUMENT-IDENTIFIER: US 5815097 A

TITLE: Method and apparatus for spatially embedded coding

DATE-ISSUED: September 29, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Schwartz; Edward L.	Sunnyvale	CA	N/A	N/A
Zandi; Ahmad	Cupertino	CA	N/A	N/A
Boros; Tibor	Stanford	CA	N/A	N/A

US-CL-CURRENT: 341/51

ABSTRACT:

The present invention provides a method and apparatus for compressing and/or decompressing data. In one embodiment, a system comprises a one-pass spatially embedded compressor and a limited bandwidth channel. The compressor comprises image data into compressed image data in one-pass. The compressor comprises an encoder and a coded data manager.

53 Claims, 34 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 26

---------- KWIC ---------

Brief Summary Text - BSTX (6):
Almost all transform domain image compression techniques are used primarily for lossy compression. This is because rounding errors in most transforms prevent their use for lossless compression. Note that the use for lossless compression implies that the compression ratio achieved for lossless compression is competitive with standard lossless techniques. Although a residue error image could be kept to make any lossy system lossless, the total "compression" achieved is usually poor. One example of a unified lossless/lossy compression system has recently been introduced as a compression with reversible embedded **wavelets** system introduced by Ricoh Corporation of Menlo Park, Calif. For more information, see Zandi, et al., "CREW: Compression With Reversible Embedded **Wavelets**", IEEE Data Compression Conference, Snowbird, Utah, pp. 212-21, March 1995 and Schwartz, et al., "Implementation of Compression with Reversible Embedded **Wavelets**", SPIE 40th Annual Meeting, vol. 2564, San Diego, Calif., July 1995.

Drawing Description Text - DRTX (11):
FIG. 9 is a block diagram of one embodiment of a **context model** for prediction coding.

Drawing Description Text - DRTX (13):
FIG. 11 illustrates a block diagram of one embodiment of a **context model** for predictive coding.

Detailed Description Text - DETX (18):
In one embodiment, both the spatially embedded compressor and the spatially embedded decompressor operate using a spatially embedded **context model** (SECM). The SECM uses a resolution reduction technique to separate the data into a low resolution "high-accuracy" image and a full resolution "low-accuracy" image. Once separated, the present invention codes the low resolution "high-accuracy" pixel losslessly, while coding the high resolution "low accuracy" pixels as accurately as possible based on the channel bandwidth available or the storage size to be used.

Detailed Description Text - DETX (28):
In the present invention, the spatially embedded compressor performs compression on either bitplanes with a hierarchical, "JBIG-like" (different templates for different phases), **context model** or on difference values using predictive coding. In one embodiment, the low- and high-accuracy images do not have to be coded using the same technique. In this case, either bitplane or predictive coding may be used. The choice of which technique to use depends, in part, on a trade-off between compression performance and cost of

implementation.

Detailed Description Text - DETX (29):
Bitplane Template **Context Models**

Detailed Description Text - DETX (30):
From a hardware perspective, **context models** can be very simple. FIG. 5A illustrates a bitplane template that provides seven bits of context (2.sup.7 =128 context bins) for binary data. FIG. 5B is a block diagram of one implementation of the template in FIG. 5A. Referring to FIG. 5B, pixels 501 are input to an optional Gray coding block 502. Gray coding is well-known in the art. One useful Gray coding operation is:

Detailed Description Text - DETX (33):
After Gray **coding, each bitplane of the data is coded as a separate binary** image. Coding "separately" means there are no dependencies so that all (or some) bitplanes may be coded in parallel. In one embodiment, the **context model** cycles through each bitplane for a given pixel before moving the template for the next pixel. When bitplanes are coded in parallel, different context bins are used for each bitplane. In this way, the probability estimation for each bitplane may remain separate.

Detailed Description Text - DETX (34):
Spatial Embedded **Context Model** Using Bitplanes

Detailed Description Text - DETX (39):
The resolution reduction of the embedded **context model** is also considered when using a bitplane template. An example set of templates is shown in FIGS. 8A and 8B. FIG. 8A illustrates bitplane templates using 20 possible pixels. These templates show available pixels for creating smaller templets, 20 is too many for most applications. FIG. 8B illustrates bitplane templates using half (10) of those 20 possible pixels as the context. In each of FIGS. 8A and 8B, a separate template is used for each of the four phases (0, 1, 2, 3) as in a "JBIG-like" system. Note that in such an embodiment, the line buffer requires multiple lines. In one embodiment, a two line buffer, with some additional memory may be used. Three and four line buffers may also be used. In some embodiments, the cost of line buffering is not important; for systems with tiling, the buffer must be able to store an entire tile. Other templates may be used. Note that to enable parallelism, different bitplanes may need to be processed in parallel, just like in a non-embedded system.

Detailed Description Text - DETX (40):
Spatial Embedded **Context Model** Using Predictive Differential Coding

Detailed Description Text - DETX (42):
There are three main differences between typical predictive coding and

typical bitplane coding. First, bitplane methods require more context bins for the same compression, and assuming each context bin requires a predetermined number of bits of memory, then the memory for larger bitplane **context models** has a high hardware cost. Second, in a bitplane **context model,** every bit is **entropy coded**. Therefore, N-bit data requires N coding operations. For predictive coding data, some of the data can be assumed to be 50% random data and simply copied to/from the compressed data without needed **entropy coding,** **thereby reducing the entropy coding** operations and allowing hardware to operate at faster speeds. Third, combining embedding with a predictive coding **context model** is more difficult if more than a few importance levels are desired. Errors cannot be allowed to propagate in a predictive coding system. In one embodiment, only the most important data is used for predictions. This avoids error propagation. If the most important data is not sufficient for making good predictions, either compression will suffer or a significantly more complicated multi-step prediction must be used.

Detailed Description Text - DETX (43):

FIG. 9 is a block diagram of one embodiment of a **context model** for coding prediction errors. Referring to FIG. 9, pixels 901 are received by local buffer 902. Local buffer 902 is coupled to line buffer 903, which provides local buffer 902 with past pixels. The output of local buffer 902 is received

by predict and determine error (e.g., interpolation block) block 904, which generates a prediction for an incoming bit and determines the error associated with the prediction with respect to the actual value of the bit. In one embodiment, the prediction is an average of two neighboring high-accuracy pixels. Other prediction methods may be used. The output of the predict and determine error block 904 includes a bit to be coded 905 and its context bin 906.

Detailed Description Text - DETX (44):
FIG. 10 illustrates one embodiment of two pixel predictors for the embedded system. Referring to FIG. 10, there are four different two pixel predictors, one for each of the four phases (0, 1, 2, and 3). Note that two lines of buffering are sufficient for the **context model,** and that this workspace may be shared with tiling or banding workspace. In one embodiment, two additional pixels, 1001 and 1002, may be used in the pixel predictor for phase 1.

Detailed Description Text - DETX (50):
FIG. 11 is a block diagram of one embodiment of a **context model** for predictive coding. Referring to FIG. 11, pixels 1101 are inputted into predictor 1102 which generates a prediction error 1103. The prediction error 1103 is converted to a sign magnitude format by sign/magnitude block 1104. The magnitude is at times referred to in the art as the

category. Table 1 illustrates one embodiment of binary magnitudes for 9 bit prediction errors. In the present invention, the bits of the pixels in addition to the sign bit and the magnitude that are required to specify the exact value of the error are referred to herein as the mantissa bits (due to the similarity to mantissas in floating point numbers). (The bits are sometimes called "make-up bits" or "tail" bits in the art). The output of sign/magnitude block 1104 is coupled to the input of binarize block 1105, which performs binarization (i.e., turns its input into binary).

Detailed Description Text - DETX (54):

In one embodiment, magnitudes are coded with a four bit number (an exponent) which is the base 2 logarithm of the magnitude. This requires only three or four **entropy coding** operations and eight context bins for the magnitude. The eight context bins are needed for the "zero-order" information that associates a unique probability estimate for each possible magnitude.

Detailed Description Text - DETX (55):

Because many of bits used to specify the sign and the mantissa are close to 50% probability (uniformly random), there is little advantage to **entropy coding** these bits. (The most significant bit of the mantissa is the only bit that may be worth coding in some systems. Using one context bin for this bit and eight

for the magnitude for a total of nine context bins and five coding operations might be a good system.) The uncoded sign and mantissa bits can be provided by shifting logic.

Detailed Description Text - DETX (57):
Referring to FIG. 12, during compression, input pixels are received by an input pixel interface 1201 and are supplied to a control model(s) 1202 which, using workspace data, provides contexts to a parallel **entropy coder** 1206. Coder 1206 generates compressed data. The compressed data manager 1207 determines which compressed data is output to a channel or memory as described above. An optional "no code" path 1205 that allows data to go uncoded to the compressed data manager 1207 for output.

Detailed Description Text - DETX (58):
During decompression, the data manager 1207 receives compressed data from a channel or memory. **Context model**(s) 1202, using the workspace data, provides contexts to the parallel **entropy coder** 1206 which decodes the compressed data. The decompressed data is output through the output pixel interface 1203. Note that if the data is not compressed, it may be sent through the optional "no code" path 1205.

Detailed Description Text - DETX (61):
In addition to specifying an embedding scheme, the **context model** of the

present invention provides context bins for magnitude bits. For nine possible magnitudes, eight context bins (as mentioned above) are sufficient to keep track of the required nine probabilities (since probabilities sum to one). Using only eight context bins provides "zero order" coding of magnitudes. Each magnitude is coded independently of its neighbors under a zero order model. (The neighbors are used for prediction only). Higher order coding can be used so that neighborhood information is used for both prediction and coding. History block 1106 in FIG. 11A provides this neighborhood information. The advantage of neighborhood information is that different probability estimates can be used for smooth regions than those used at edges, which allows better compression.

Detailed Description Paragraph Table - DETL (3):

For each pixel in image decorrelate based on causal data (e.g., prediction, Gray coding, . . .) (optional) binarize for each bit in binarization generate context: condition bit on causal information determine importance level of bit **entropy code** (or optionally, output without coding) output any coded data created to a coded data manager

Detailed Description Paragraph Table - DETL (4):

For each

pixel in image for each bit
in binarization generate context: condition bit on casual information
determine importance level of bit fetch any coded data needed from the coded
data manager, **entropy decode (or optionally, output without coding**) if (not
lossless) reconstruct inverse binarize inverse decorrelate based on casual
data (optional)

Claims Text - CLTX (3):

an encoder having a spatially embedded **context model,** wherein the encoder
operates in the spatial domain to encode first image data, and

Claims Text - CLTX (21):

an encoder having a spatially embedded **context model** to perform resolution
reduction to obtain a low resolution, high-accuracy image and high resolution,
low-accuracy image, and

Claims Text - CLTX (23):

7. The compressor defined in claim 6 wherein the **context model** performs
resolution reduction by subsampling.

Claims Text - CLTX (24):

8. The compressor defined in claim 6 wherein the **context model** performs
resolution reduction using two dimensional subsampling.

Claims Text - CLTX (25):
9. The compressor defined in claim 6 wherein the **context model** generates contexts using bitplane templates.

Claims Text - CLTX (27):
an encoder having a spatially embedded **context model** to perform predictive coding, and

Claims Text - CLTX (29):
11. The compressor defined in claim 10 wherein the encoder codes prediction errors from the **context model**.

Claims Text - CLTX (30):
12. The compressor defined in claim 11 wherein the **context model** performs prediction using interpolation.

Claims Text - CLTX (31):
13. The compressor defined in claim 6 wherein the **context model** performs Gray coding for bitplane coding.

Claims Text - CLTX (34):
an encoder having a spatially embedded **context model,** wherein the encoder operates in the spatial domain to encode first image data, and

Claims Text - CLTX (54):
an encoder having a spatially embedded **context**

model to perform resolution
reduction to obtain a low resolution, high-accuracy
image and high resolution,
low-accuracy image, and

Claims Text - CLTX (57):
20. The system defined in claim 19 wherein the **context model** performs
resolution reduction by subsampling.

Claims Text - CLTX (58):
21. The system defined in claim 19 wherein the **context model** performs
resolution reduction using two dimensional
subsampling.

Claims Text - CLTX (59):
22. The system defined in claim 19 wherein the **context model** generates
contexts using bitplane templates.

Claims Text - CLTX (62):
an encoder comprising a spatially embedded **context model** to perform
predictive coding, and

Claims Text - CLTX (65):
24. The system defined in claim 23 wherein the encoder codes prediction
errors from the **context model**.

Claims Text - CLTX (66):
25. The system defined in claim 24 wherein the **context model** performs

prediction using interpolation.

Claims Text - CLTX (67):
26. The system defined in claim 19 wherein the **context model** performs Gray coding for bitplane coding.

Claims Text - CLTX (74):
entropy coding said each bit based on the importance level; and

Claims Text - CLTX (83):
entropy coding said each bit; and

Claims Text - CLTX (93):
entropy coding said each bit; and

Claims Text - CLTX (101):
entropy coding said each bit; and

Claims Text - CLTX (109):
entropy coding said each bit; and

Claims Text - CLTX (117):
entropy coding said each bit;

Claims Text - CLTX (133):
entropy coding said each bit; and

Claims Text - CLTX (140):
(a) generating a context, using a spatially

embedded **context model** for each
bit in a binarization by conditioning said each bit
on causal information;

Claims Text - CLTX (199):
means for **entropy coding** said each bit; and

Claims Text - CLTX (200):
means for outputting any coded data to a coded
data manager, wherein the
means for generating, means for determining, means
for separating, means for
entropy coding and means for outputting operate on
each bit in the
linearization.

Other Reference Publication - OREF (4):
Ahmed Zandi, Martin Boliek, Edward L. Schwartz,
Alexander Keith,
"Compression with Reversible Embedded **Wavelets** with
an Enhanced Binary Mode",
submitted to the IEEE on Acoustics, Speech, and
Signal Processing, Atlanta,
Georgia, May 7, 1996, p. 4.

US-PAT-NO: 5818877

DOCUMENT-IDENTIFIER: US 5818877 A
See image for Certificate of Correction

TITLE: Method for reducing storage requirements for grouped data values

DATE-ISSUED: October 6, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Tsai; Min-Jen	Los Angeles	CA	N/A	N/A
Villasenor; John D.	Santa Monica	CA	N/A	N/A
Chen; Feng	Irvine	CA	N/A	N/A

US-CL-CURRENT: 375/241, 341/106 , 341/55 , 341/67 , 358/426.13 , 375/253 , 382/245

ABSTRACT:

A method generates a reduced number of values representing a sequence of grouped data values and partitions the reduced number of values by first mapping data values into groups of symbols and then partitioning the resulting stream of symbols. The digits representing the first data value in each group are replaced with symbols from a first alphabet. The most significant digit of the second data value in each group and the sign of

that second data value are
also represented by a symbol from the first symbol set, while the remaining
significant bits of the second data value in each group are represented by
symbols from a second symbol set. A stream of symbols which represent a
sequence of grouped data values is partitioned into first partition symbol
groups and second partition symbol groups. Each first partition symbol group
comprises the symbols representing the first data value in each group and also
the symbol representing the least significant bit of the second data value
which follows. Each second partition symbol group comprises the symbols
representing all digits of a second data value excluding the symbol which
represented the least significant digits.

13 Claims, 38 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 20

---------- KWIC ---------

Brief Summary Text - BSTX (10):

Irreversible compression methods typically comprise several sequential
steps, some of which are irreversible, and others of which are reversible.
Many compression methods use the sequence of steps: transformation,
quantization, run-length **coding, and entropy coding**.

Brief Summary Text - BSTX (11):
The transformation step gives an alternative representation of the input data, and is reversible provided that the transform arithmetic has sufficient precision. In the quantization step, transformed data values are approximated by rounding to the nearest member of a set of permitted quantizer output values. This step is irreversible because each quantizer output value only approximates an associated input value. The subsequent steps of run-length **coding, and then of entropy coding** the signals are both reversible.

Brief Summary Text - BSTX (14):
More recently, coding methods using discrete **wavelet** transforms (DWT) have also been developed and included in industry or government sponsored standardization efforts. For example, the Federal Bureau of Investigation has recently formally adopted the **Wavelet** Scalar Quantization (WSQ) method, which is based on the **wavelet** transform instead of on the DCT. In many DWT-based methods, including WSQ, after the **wavelet** transform is taken, quantization, run-length **coding, and entropy coding** are used.

Brief Summary Text - BSTX (16):
An important step in all of the above coding methods is the creation of a list of integers representing quantized data. This list can be described in

terms of the size of runs of consecutive zeros, and the value or level of the nonzero coefficient that terminates each run of zeros. The numbers in this list are defined as (run, level) pairs. In particular, the way in which this information is mapped for subsequent **entropy coding** plays a critical role in the performance of the compression method. Previous methods have typically used a separate symbol to represent each(run, level) pair. The frequency of occurrence of all the symbols is tabulated, and is used in an **entropy coding** method in which frequently occurring symbols are represented using fewer bits.

Brief Summary Text - BSTX (22):

In one application of this invention, a new mapping of a list of (run, level) pairs into symbols is used. This invention takes advantage of the fact that the unsigned binary representation of any number, considering only significant bits, is a word that begins with "1", and that if some other means is used to represent the number of digits of the **binary word, then this "1" does not need to be explicitly coded**. In the present invention, binary representations for runs and levels are ordered from least significant bit (LSB) to most significant bit (MSB), where the MSB is always 1. In the majority of run values and level values, it is possible to explicitly omit encoding the MSB by using another means of terminating the word.

Brief Summary Text - BSTX (26):
Furthermore, the presence of two contexts is used to increase the performance of the **entropy coding** step that follows the mapping to symbols. Separate probability tables are developed for each context, and these probability tables are used to design optimized **entropy codes**. If the statistics of the symbol occurrences in the two contexts changes, the **entropy codes** are adapted.

Detailed Description Text - DETX (3):
The digital data at input terminal 102 is input to a transform processor 104. The transform processor 104 generates transformed data which provide an alternative representation of the input data. The transform processor 104 can use one of a number of known transforms including, for example, Fourier, discrete cosine, or **wavelet**. The transform processor 104 may be a software configured computer or a custom firmware or hardware circuit for performing the desired transform.

Detailed Description Text - DETX (9):
The output of the symbol mapper 110 is coupled to a context analyzer 112. The context analyzer, too, may be a software configured computer or a custom firmware or hardware circuit. The context analyzer 112 advantageously partitions a sequence of run and level words generated by the symbol mapper 110

into two different contexts. As will be described in more detail herein, partitioning the run and level words into differing word-oriented contexts allows greater compression to be achieved by the later step of **entropy coding**.

Detailed Description Text - DETX (10):

The present invention preferably includes an optional reorderer 114 which reorders symbols within the run and level words prior to the step of **entropy coding**. As will be described in more detail below, symbol reordering can result in still greater compression ratios when run/level coded data contain large percentages of high run or level values.

Detailed Description Text - DETX (11):

The reordered data are input to an **entropy coder** 116, which preferably comprises an adaptive arithmetic coder, but may alternatively be a conventional Huffman **coder or other known entropy coder**. The preferred adaptive arithmetic coder is responsive to both run-oriented and level-oriented contexts, and can thereby achieve greater compression. Adaptive arithmetic coding is known in the art. See, for example, I. H. Witten, R. Neal, and J. G. Cleary, "Arithmetic Coding For Data Compression," Communications of the ACM, v. 30, pp. 520-540 (1987); Y. H. Kim and J. W. Modestino, "Adaptive **Entropy Coded** Subband Coding Of Images," IEEE Trans. Image Processing, v. 1, pp. 31-48 (1992). The preferred **entropy coder** 116 switches coding

contexts upon receiving from the context analyzer 112 (1) a group of symbols associated with a particular context and (2) a command to select the particular coding context. **Entropy coding** (such as arithmetic coding) is a method of compressing data by minimizing the number of bits used to represent the most frequently occurring data values in input data. The **entropy coder** 116 may be a program-enabled computer, or a firmware or hardware circuit. The **entropy coder** 116 produces a compressed bit stream, which represents the final result of the data compression system of FIG. 1.

Detailed Description Text - DETX (12):

FIG. 2 is a block diagram of a digital decompression system. Compressed bit stream data at input terminal 202 are input to an **entropy decoder 204 which reverses the entropy coding performed by the entropy coder** 116. The entropy decoder 204 outputs symbols which comprise run or level words. The decompression system of FIG. 2 preferably includes a reverse symbol reorderer 206 if corresponding symbol reordering was performed during compression. As further described below, a context analyzer receives decoded symbols from the entropy decoder 204 and controls context-sensitive **entropy decoding in cases where entropy coding** was performed in a context-sensitive manner. The output of the entropy decoder 204, as assisted by the context analyzer 208, is a stream of symbols which is input to an inverse

symbol mapper 120.

Detailed Description Text - DETX (14):
The quantized integer data are input to an inverse quantizer 214. The inverse quantizer 214 replaces each integer with a floating-point value approximating that integer's original pre-quantization, floating-point value. The output of the inverse quantizer thus represents an approximation of transformed data, the data therefore being at least slightly different from the original transformed data output by the transform processor 104. The approximated transformed data are input to an inverse transform processor 216. The inverse transform processor 216 reverses the transform (Fourier, discrete cosine, **wavelet,** or other) performed by the transform processor 104. The output of the inverse transform processor 216 is an approximation of the original data input to the data compression system of FIG. 1.

Detailed Description Text - DETX (22):
It is also desirable to perform the mapping such that the symbols comprising words appear with unequal probability. This will lead to greater efficiency in the **entropy coding performed by the entropy coder** 116. Given knowledge that smaller run values are more probable than larger run values in certain input data, Symbol Map 2 504 illustrates a particular mapping that will cause the symbol "+" to occur more frequently than "-".

Symbol Map 2 thus is arranged such that, among any group of words having the same number of symbols, words associated with the smallest run values contain the most "+" symbols. Thus, where run values have a higher probability of being small, Symbol Map 2 504 can be expected to cause the symbol mapper 110 to output run words in which the "+" symbol will appear with greater frequency than the "-" symbol.

Detailed Description Text - DETX (29):

Although FIG. 6 demonstrates a binary value correspondence between level values and the words to which they are mapped, the symbol mapper 110 of the present invention could employ a symbol map (or look up table) to alternatively assign words to level values that do not have a binary correspondence. FIG. 7 illustrates two symbol maps which could be used to assign words to level values. Symbol Map 1 702 represents the symbol mapping strategy of FIG. 6 wherein words are assigned to level values in a binary ascending order. Symbol Map 2 704, however, represents an alternate map for associating words with level values. As explained above with respect to symbol mapping for run values, additional efficiency in **entropy coding** can be gained when input symbols appear with unequal probability. Within any group of level words having the same length, Symbol Map 2 arranges the words so that the symbol "0" occurs more frequently in words associated with small level values. Thus, in

the case where run/level coded data contain a large percentage of low level values, better compression ratios could be expected by using Symbol Map 2 704 instead of Symbol Map 1 702.

Detailed Description Text - DETX (40):

On the other hand, if the present invention is applied in a **wavelet** transform system in which the data set is large enough to permit run values of many hundreds of thousands or greater, then a logical construction method is preferred. This occurs because use of a symbol map in this case would require a look up table with many hundreds of thousands of entries, which would be expensive using current hardware memory devices. However, it will be understood that as memory devices increase in capacity in the coming years, longer and longer look up tables will become more practical to implement.

Detailed Description Text - DETX (51):

Entropy coding is preferably performed by an arithmetic coder 116 that is not only adaptive, but which can also code data in different contexts. The present invention will obtain superior compression levels using an arithmetic coder 116 (FIG. 11) capable of maintaining two **context models** during coding--one **context model** 1104 for coding a first symbol stream and a second **context model** 1106 for coding a second symbol stream, where the first and second symbol streams have different symbol

frequencies. Adaptive arithmetic coders which adjust a coding model to an input stream as input data are being coded are known in the art and will not be described further.

Detailed Description Text - DETX (52):

The context analyzer 112 of the present invention, upon sending to the arithmetic coder 116 a group of symbols comprising a run word and the LSB symbol of the following level word, issues a select context command 1102 to the arithmetic coder 116. The select context command 1102 causes the arithmetic coder 116 to use the coding **context model** 1104 which it is adapting to arithmetically code groups of symbols which comprise a run word and LSB symbol.

Detailed Description Text - DETX (53):

Similarly, the context analyzer 112, upon sending to the arithmetic coder 116 a group of symbols comprising a level word with no LSB, issues a second select context command 1102 to the arithmetic coder. This second select context command 1102 causes the arithmetic coder 116 to use the **context model** 1106 it is adapting to code groups of symbols consisting of a level word having no LSB.

Detailed Description Text - DETX (54):

By partitioning the input symbol stream into two types of symbol groups, the invention takes advantage of the different symbol

frequencies associated with each type of symbol group. The context-sensitive arithmetic coder 116 can thus achieve greater compression by coding symbols from each of the two groups using two different adaptive **context models** (1104, 1106). The output of the arithmetic coder 116 is a compressed bit stream 1108.

Detailed Description Text - DETX (62):
Bit-plane reordering, although introducing greater complexity in the compression process, will cause further inequality in the distribution of symbols within the symbol groups being reordered, and, as discussed above, will facilitate improved compression by the later step of **entropy coding**. Other types of reordering are possible and the present invention is not limited to the bit-plane reordering described herein.

Detailed Description Text - DETX (64):
FIG. 15 is a block diagram which illustrates the steps of inputting compressed bitstream data 1108 to the arithmetic decoder 204 and decoding the bitstream data 1108 into a symbol stream 1502 that is eventually input to an inverse symbol mapper. The arithmetic decoder 204 is capable of decoding the input bitstream 1108 using two different contexts, and is thereby capable of decoding the bitstream 1108 produced by the arithmetic coder 116 described in relation to FIG. 11. The arithmetic decoder 204 is initialized to decode

bitstream data using a first **context model** 1504 appropriate for decoding bitstream data which represent symbols comprising symbol groups from a location list 1204 (groups of symbols comprising a run word and a LSB symbol from a following level word). Thus, when the compressed bitstream 1108 is input to the arithmetic decoder 204, the arithmetic decoder 204 outputs a symbol stream 1502 containing symbols from a location list 1204 symbol group. As the symbols 1502 are output one-by-one by the arithmetic decoder 204, these symbols are input to the context analyzer 1506. The context analyzer 1506 examines the input symbols 1502 one-by-one.

Detailed Description Text - DETX (65):

As the context analyzer 1506 examines the first few input symbols, it detects the termination of a location list 1204 symbol group, and then issues a select context command 1208 to the arithmetic decoder 204. The arithmetic decoder 204 then uses a **context model** 1510 appropriate to decode bitstream data representing a symbol group from a level list 1206. The arithmetic decoder 204 then begins to output symbols representing a level list 1206 symbol group. When the context analyzer 1506 recognizes the termination of a level list 1206 symbol group, it issues a select context command 1208 to the arithmetic decoder 204 to cause the arithmetic decoder 204 to again alter the **context model** used to decode the bitstream data 1108. Thus, the context analyzer 1506 controls

the context of arithmetic decoding until the bitstream input to the arithmetic decoder 204 is exhausted. The context analyzer 1506 outputs the symbol stream 1504 unaltered for input to an inverse symbol mapper.

Detailed Description Text - DETX (68):

Because the data of each bitstream (1316, 1318) represent only one context, the arithmetic decoders (1602, 1604) need not be capable of decoding in multiple **context models**. The arithmetic decoder 1602 outputs a sequence of symbols representing symbols in the location list 1308, and the arithmetic decoder 1604 outputs symbols representing symbols in the level list 1312 which have been word-to-bit-plane reordered.

DERWENT-ACC-NO: 2000-173981

DERWENT-WEEK: 200370

TITLE: Reversible embedded wavelet system implementation method for lossless and lossy encoding and decoding of data in compression and decompression systems

INVENTOR: SCHWARTZ, E L

PRIORITY-DATA: 1997US-0847074 (May 1, 1997)

PATENT-FAMILY:

PUB-NO		PUB-DATE
LANGUAGE	PAGES	MAIN-IPC
GB 2341035 A		March 1, 2000
N/A	166	H04N 007/26
GB 2341035 B		November 15, 2000
N/A	000	H04N 007/26

INT-CL (IPC): H04N007/26

ABSTRACTED-PUB-NO: GB 2341035A

BASIC-ABSTRACT:

NOVELTY - The method comprises steps of: dividing a coefficient into **most important** and less important **data**; sending **most important data to a context model** for coding immediately in coefficient order;

storing the less important
data and several signaling **bits** in memory; and
after coding **most important data**
of all coefficients in the set of coefficients,
coding the less important **data**
and embedding by order based, in part, on the
number of signaling **bits**.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(1) an apparatus for compressing an image

(2) an apparatus for coding information

(3) a method for m-ary coding of information

(4) an integrated circuit (IC) chip

(5) a decoder

(6) a context model

(7) a method for performing compression

(8) a system, and

(9) a method for processing a least important portion of data.

USE - For lossless and lossy encoding and decoding of data in
compression/decompression systems.

ADVANTAGE - Provides either excellent lossless or lossy compression as required
by image characteristics and the bursty nature of the hard disk.

DESCRIPTION OF DRAWING(S) - The drawing shows the

context dependent relationships. Children are conditioned on their parents.

ABSTRACTED-PUB-NO: GB 2341035B

EQUIVALENT-ABSTRACTS:

NOVELTY - The method comprises steps of: dividing a coefficient into **most important** and less important **data**; sending **most important data to a context model** for coding immediately in coefficient order; storing the less important **data** and several signaling **bits** in memory; and after coding **most important data** of all coefficients in the set of coefficients, coding the less important **data** and embedding by order based, in part, on the number of signaling **bits**.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(1) an apparatus for compressing an image

(2) an apparatus for coding information

(3) a method for m-ary coding of information

(4) an integrated circuit (IC) chip

(5) a decoder

(6) a context model

(7) a method for performing compression

(8) a system, and

(9) a method for processing a least important portion of data.

USE - For lossless and lossy encoding and decoding of data in
compression/decompression systems.

ADVANTAGE - Provides either excellent lossless or lossy compression as required
by image characteristics and the bursty nature of the hard disk.

DESCRIPTION OF DRAWING(S) - The drawing shows the context dependent
relationships. Children are conditioned on their parents.

---------- KWIC ---------

Basic Abstract Text - ABTX (1):

NOVELTY - The method comprises steps of: dividing a coefficient into **most important** and less important **data**; sending **most important data to a context model** for coding immediately in coefficient order; storing the less important **data** and several signaling **bits** in memory; and after coding **most important data** of all coefficients in the set of coefficients, coding the less important **data** and embedding by order based, in part, on the number of signaling **bits**.

Equivalent Abstract Text - ABEQ (1):

NOVELTY - The method comprises steps of: dividing a coefficient into **most important** and less important **data**; sending **most**

important data to a context model for coding immediately in coefficient order; storing the less important **data** and several signaling **bits** in memory; and after coding **most important data** of all coefficients in the set of coefficients, coding the less important **data** and embedding by order based, in part, on the number of signaling **bits**.

US-PAT-NO: 5381145

DOCUMENT-IDENTIFIER: US 5381145 A

TITLE: Method and apparatus for parallel decoding and encoding of data

DATE-ISSUED: January 10, 1995

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Allen; James D.	Castro Valley	CA	N/A	N/A
Boliek; Matin	Palo Alto	CA	N/A	N/A
Schwartz; Edward L.	Sunnyvale	CA	N/A	N/A

US-CL-CURRENT: 341/107, 341/51 , 358/1.9 , 375/240.1

ABSTRACT:

The present invention provides a method and apparatus for encoding and decoding data in parallel. The present invention provides a system for decompressing a data stream having multiple codewords. The system includes an input channel that receives the data stream. The system also includes a decoder which decodes each bit of the data stream, wherein at least two of the codewords in the data stream are decoded at the same time, such that the data stream is decoded in parallel.

84 Claims, 28 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 23

---------- KWIC ---------

Brief Summary Text - BSTX (9):

FIG. 1 shows a block diagram of a prior art compression/decompression system using a **binary entropy coder**. For coding, data is input into **context model** (CM) 101. CM 101 translates the input data into a set or sequence of binary decisions and provides the context bin for each decision. Both the sequence of binary decisions and their associated context bins are output from CM 101 to the probability estimation module (PEM) 102. PEM 102 receives each context bin and generates a probability estimate for each binary decision. The actual probability estimate is typically represented by a class, referred to as PClass. Each PClass is used for a range of probabilities. PEM 102 also determines whether the binary decision (result) is or is not in its more probable state (i.e., whether the decision corresponds to the MPS). The bit-stream generator (BG) module 103 receives the probability estimate (i.e., the PClass) and the determination of whether or not the binary decision was likely as inputs. In response, BG module 103 produces a compressed data

stream, outputting zero or more bits, to represent the original input data.

Brief Summary Text - BSTX (11):

The **context model** is typically application specific. Since any type of data can be reduced to bits, a **binary entropy coder** with the proper **context model** can be used for any data. An example of a **context model** is given by the JBIG Standard (ISO/IEC International Standard, "Coded Representation of Picture and Audio Information-Progressive Bi-level Image Compression Standard").

Brief Summary Text - BSTX (15):

One problem with decoders using **binary entropy codes,** such as IBM's Q-coder and the B-coder, is that they are slow, even in hardware implementation. Their operation requires a single large, slow feedback Iccp. To restate the decoding process, the **context model** uses past decoded data to produce a context. The probability estimation module uses the context to produce a probability class. The bit-stream generator uses the probability class and the compressed data to determine if the next bit is the likely or unlikely result. The probability estimation module uses the likely/unlikely result to produce a result bit (and to update the probability estimate for the context). The result bit is used by the **context model** to update its history of past data. All of these steps are required for decoding a single bit. Because the **context model** must wait for

the result bit to update its history before it can provide the next context, the decoding of the next bit must wait. Therefore, parallel decoding of a single coded data stream does not occur in the prior art. It is desirable to decode data in parallel in order to increase the speed at which compressed data is decoded.

Detailed Description Text - DETX (14):

Binary entropy codes use multiple context bins and multiple probability estimate states (classes). The method and apparatus of the present invention allow parallelism to process context bins in parallel or handle probability classes in parallel. When processing context bins in parallel, a probability estimation module is associated with each bit-stream generator. It indicates to its associated bit-stream generator which code to utilize to produce a data stream from the input codewords. An example of such a system is shown in FIG. 2C where coded data is coupled as an input to channel control 221. Channel control 221 receives the coded data and directs coded data to each of multiple bit-stream generators (e.g., BG 222, BG 223, BG 224, etc.). Each of the bit-stream generators is coupled to receive the coded data and provides the result of whether the codeword was in its most likely state or not to its associated probability estimation module, in response to the probability class provided by the probability estimation module (PEM). PEMs 225, 226, 227, etc.

are coupled to BGs 222, 223, 224, etc. respectively. Each bit-stream generators can operate independently of the others because it is decoding coded data which always has the same context bin. The **context model** 228 is coupled to each of the probability estimation modules and selects the probability estimation modules to obtain the decoded data in the determination order of the application. In this manner, decoded data is produced by processing context bins in parallel.

Detailed Description Text - DETX (76):

The present invention uses non-interleaved channel or multi-channel **coders in parallel to increase the coding speed in binary entropy coders**. Decoders for **binary entropy coders,** such as IBM's Q-coders and B-coders, require a single large feedback loop between the decoder and the **context model,** as described previously. By using proper coded data transmission and multiple decoders, the **context model** feedback loop is isolated. By isolating the **context model** feedback loop, the **context model** may be reduced to selecting an already decoded result.

US-PAT-NO: 5471207

DOCUMENT-IDENTIFIER: US 5471207 A

TITLE: Compression of palettized images and binarization for bitwise coding of M-ary alphabets therefor

DATE-ISSUED: November 28, 1995

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Zandi; Ahmad	Cupertino	CA	N/A	N/A
Stork; David G.	Sanford	CA	N/A	N/A
Allen; James	Mountain View	CA	N/A	N/A

US-CL-CURRENT: 341/107, 341/51 , 382/232

ABSTRACT:

The invention provides an improved method and apparatus for compression of palettized images. Input symbols in an M-ary alphabet are binarized based on a context model of the input data, where the binarization is selected to provide good compression by a binary encoder. The particular binarization is determined from a reindexing table which maps each input symbol to a number of binary values. The mapping is determined from the images to be compressed, and is typically transmitted with the compressed images

as overhead. The mapping
is a local minimum of the bitwise entropy of the binarization. With or without
reindexing the input, the symbols can be converted compressed in parallel, with
the bits of the input symbols buffered and reordered as necessary to ensure
that bits needed for context of a bit being decoded are available before the
decompressor decodes the bit being decoded. The decompressor includes a means
for performing the opposite reordering such that the output of the decompressor
is the same as the input to the compressor.

24 Claims, 14 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 12

---------- KWIC ---------

Brief Summary Text - BSTX (18):

In one embodiment of the present invention, input symbols are symbols in an M-ary alphabet and are binarized based on a **context model** of the input data, where the binarization is selected to provide good compression by a **binary coder**. An encoder converts input symbols into codewords, while a decoder converts the codewords into the input symbols. The particular binarization is determined from a reindexing table which maps each input symbol to a number of binary values. The mapping is determined from the images to be compressed, and

is typically transmitted with the compressed images as overhead.

Detailed Description Text - DETX (2):

The description below is divided into several sections. The first section describes **entropy coding**. The second section describes the use of reindexing to improve compression when using **entropy coding**. The third section describes how **context modelling** can improve **entropy coding** performance by more accurately estimating symbol probabilities. **Context modelling** can be used with or without reindexing. The fourth section describes how parallel compressors and decompressors are used to improve data rates. Finally, the fifth section describes how buffering is used to facilitate parallel implementations and still maintain causality to allow data to be decompressed in parallel.

, however the **context models** described herein are an improvement over the prior art, and hence are the objects of the prese